

**REMARKS**

At the time of the office action, Claims 1, 3, 5, 7 to 10, 13, and 15 to 24 were pending in the application. Claims 1, 3, 5, 7 to 10, 13, and 15 to 24 were rejected as obvious.

Claims 1, 3, 5, 8 to 10, 13, 15, 16, 18 to 21, and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over by U.S. Patent No. 6,012,098, hereinafter referred to as Bayeh, in view of "Servlet Essentials," hereinafter referred to as Zeiger.

**Requirements for a Prima Facie Obviousness Rejection**

To make a prima facie obviousness rejection, the MPEP directs:

**BASIC CONSIDERATIONS WHICH APPLY TO OBVIOUSNESS REJECTIONS**

When applying 35 U.S.C. 103, the following tenets of patent law must be adhered to:

- (A) The claimed invention must be considered as a whole;
- (B) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination;
- (C) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and
- (D) Reasonable expectation of success is the standard with which obviousness is determined.

MPEP § 2141, 8th Ed., Rev. 2, p. 2100-120 (May 2004). It is noted that this directive stated "the following tenets . . . **must be adhered to.**" Accordingly, the failure of the Examiner to adhere to any one of these tenets means that a prima facie obviousness rejection has not been made.

The rejection failed to adhere to multiple of these tenets and so a prima facie obviousness rejection has not been made. As demonstrated more completely below, the claimed invention has not been considered as a whole; the references have not

been considered as a whole; and the references do not suggest the desirability of making the combination. Pieces of the references have been extracted and selectively interpreted in view of Applicants' claims.

Applicants respectfully traverse the obviousness rejection of Claim 1. As previously noted:

Bayeh describes a servlet chain that has a data servlet and a rendering servlet. Apparently, this is what is being cited as a plurality of partial filter adapters.

However, each of the servlets, as described by Bayeh, does not have an interface as recited in Claim 1. This is because Bayeh is solving a different problem isolating "data retrieval from data presentation formatting." Bayeh taught:

The role of the data servlet is only to retrieve data from a database 88': it does no presentation formatting of that retrieved data. The data servlet 83 receives the search request 80', queries a database 88' using database query statements 86' appropriate to the particular database, and receives the query results 90'. At that point, the data retrieval function of the data servlet 83 is complete.

Bayeh, Col. 8, lines 7 to 12.

Bayeh further taught:

At Step 240, the data servlet processes the client request. The request will typically require retrieving data from some database available to the data servlet. The data servlet will format a database query request, using an appropriate query language that will depend on the type of database on which the relevant data is stored. Database query languages are well known in the art, as are techniques for determining which language is required and how to format queries in a particular language. The creation of the query request, as well as sending the request to the database and receiving data satisfying that request, do not form part of the inventive concepts of the present invention. Techniques are used which are well known in the art.

Bayeh, Col. 10, lines 46 to 58.

Thus, the functions described in these quotes from Bayeh fail to teach that the data servlet includes any type of interface for receiving input data as recited in Claim 1. The rejection has cited no teaching that the data servlet is used for other than retrieving data using an appropriate query language and then formatting that data specifically for the next servlet. Bayeh provides no detail on whether there is an interface and teaches only that the input data is obtained using known database retrieval techniques. Moreover, there is no teaching that any interface associated with database retrieval techniques is also utilized in the rendering servlet to receive input data.

To address this defect in Bayeh, the rejection stated:

Bayeh does not explicitly disclose the exact manner in which the servlets communicate with each other and explicitly suggests that one would look to the prior art for guidance on such matters (col. 9, ll. 62-63 and col 10, ll 14-15).

Applicants respectfully submit that this statement mischaracterizes the teaching of Bayeh and fails to consider Bayeh as a whole. In the cited sections, Bayeh is not describing or considering how servlets communicate. Specifically, Bayeh, Col. 9, lines 62 and 63 stated:

Servlet aliasing is known in the art, as are techniques for creating comma-separated list.

This portion of Bayeh teaches nothing about servlet communication and instead described "servlet aliasing" and "creating comma-separated list." Further, taken in context, Bayeh taught:

When a servlet chain is to be used for processing a client request, there are two ways currently known in the art to trigger the chained processing: (1) "servlet aliasing" or (2) Mime types. Either technique can be used to implement the present invention.

Bayeh, Col. 9, lines 39 to 42.

Thus, Bayeh expressly taught that "servlet aliasing" or "Mime types" triggers the chained processing. Triggering chained processing fails to teach going to the prior art to learn about "interfaces."

Both of the sections cited in the rejection are related to these two techniques for triggering chained processing and not anything that describes or suggests communications between servlets. Specifically, with respect to servlet aliasing, Bayeh taught:

Servlet aliasing is a technique whereby a list of servlets is predefined, with the identifying name of each servlet specified in a comma-separated list, in the order in which the servlets are to be invoked. The name of the data servlet would typically be the first-named servlet in this list, since this data servlet needs to retrieve the data satisfying the user's request. However, it is possible that some type of pre-processing of the user's request might be required, in which case the name of the servlet performing the pre-processing would appear in the comma-separated list before the name of the data servlet. The last name in the comma-separated list is the name of the rendering servlet. This comma-separated list is defined as being associated with a specific Uniform Resource Locator ("URL"), where a URL is analogous to the address of the server computer where the client request should be sent. **Servlet aliasing is known in the art, as are techniques for creating the comma-separated list.** (Emphasis indicates portion cited in the rejection.)

Bayeh, Col. 9, lines 46 to 63.

Thus, Bayeh expressly described "servlet aliasing" and a comma-separated list and then to avoid going into further detail simply stated the techniques for these two items were known to those of skill in the art. Thus, this portion of Bayeh taken as a whole fails to suggest or disclose anything concerning how the servlets in the comma-separated list communicate. To the extent that the rejection implies

otherwise is evidence that Bayeh has not been considered as a whole and evidence that Bayeh is being mischaracterized.

Similarly, Bayeh, Col. 10, lines 14 and 15 are at the end of a paragraph describing MIME types, specifically:

Mime is an abbreviation for Multipurpose Internet Mail Extensions, and refers to a standardized method of allowing multiple types of data to be intermixed in data streams. A mime type is used to identify what type of data is contained in the data stream. A configuration file identifies particular mime types with the name of the executable process to be invoked when a data stream having that mime type is received. Chaining is implemented by creating an entry in the "mimeservlets.properties" file to specify the servlet name that will be invoked for a particular mime type. As each servlet in the chain creates its output stream, it associates a mime type with that output. Typically, a data servlet of the present invention will create its output with a mime type of "text/xml", and the output of the rendering servlet will have a mime type of "text/html". By associating the mime type "text/xml" with the name of a rendering servlet in the mimeservlets.properties file, the rendering servlet is automatically invoked to process the output created by the data servlet. **These techniques, and the manner of implementing them, are known in the art.** (Emphasis indicates portion cited in the rejection.)

Again, the techniques being referred to are creating a file based upon MIME types for associating a data servlet with a rendering servlet, as Bayeh stated "By associating the mime type "text/xml" with the name of a rendering servlet in the mimeservlets.properties file, the rendering servlet is automatically invoked to process the output created by the data servlet." It is these techniques and the manner of implementing these techniques that was stated as known. Again, this portion of Bayeh taken as a whole fails to suggest or disclose anything concerning how the servlets in the file communicate.

Nevertheless, the rejection stated:

Zeiger discloses that a preferred way for servlets to communicate with other is to use a Java construct called an interface, which is used to ensure that different data types can communicate with each other (p. 30, para 3).

Zeiger does not mention data types or different data types, and instead refers to an object-orientated programming construct "classes" and in particular classes that define servlets. Accordingly, the equating of a servlet with a data type is not supported by the reference. The rejection has cited no teaching of data types in Ziegler and so this portion of the rejection is not supported by Zeigler.

Further, the problem begin addressed by Zeigler is described as "This section shows how to call a method of another servlet," at page 29 of Ziegler. Calling a method of another servlet using an interface teaches or suggests nothing about how the method is implemented or about the characteristics of either the interface or the servlet making the call to the other servlet.

A method requires definition before any statements can be made about the characteristics of the method. Therefore, this is further evidence that the characterizations in the rejection are not supported by Zeiger. The method taught by Zeiger is "method public void bar()" All this tells us is that the name of the method is "bar". Zeiger provides no detail as to what the method bar does or how it performs. Consequently, Zeiger provides no information on the functionality or characteristics of the interface other than the interfaces includes a method having this name.

At most Zeiger teaches how an interface loaded by a system loader can be shared by all servlets. The methods in such an interface and the implementation of such methods are not described by Zeiger, and so yet again this demonstrates that the characterization of Zeiger in the rejection is not well founded.

The rejection mischaracterizes the teaching of Zeiger by stating:

Therefore it is a generic source and target data format independent interface.

Since Zeiger provides no detail on the interface other than it includes a method named bar, there is no basis, when the reference is considered as a whole, to support any characterization of the functionality provided by such an interface. Thus, the above conclusion in the rejection is neither suggested nor taught by Zeiger. If the rejection is continued, the Examiner is respectfully requested to cite with specificity the portions of Zeiger that support each of the conclusions made in the rejection.

Further, the rejection has failed to explain how such an interface would be incorporated into either the comma separated list used in servlet aliasing of Bayeh, or the file used with the MIME types, which were the items cited by Bayeh as being known to those of skill in the art. Techniques for triggering chained processing are at a different functional level from a servlet calling a method in another servlet via an interface and so the modification would change the principles of operation of Bayeh for triggering the chained processing. This also is sufficient to overcome the rejection.

Finally, the rejection stated:

to use the interfaces of Ziegler to enable the intra-servlet communication necessary for Bayeh.

Bayeh taught how the data servlet and the rendering servlet communicated via an XML stream. Further, the statements of Bayeh as to looking to the prior art relied upon in the rejection were not directed to communications between servlets as characterized by the rejection, but rather to techniques for triggering chained processing as discussed

above. Thus, the motivation is unrelated to the portions cited in Bayeh.

In addition, if the interface were used, Ziegler fails to teach anything specific about such an interface. In particular, there is no showing that the same interface would be used in both the data and rendering servlets of Bayeh.

Claim 1 recites:

each partial filter adapter includes a generic source and target data formats independent interface and said generic source and target data formats independent interface is for receiving input data by each partial filter adapter independent of an underlying data format of said input data

General knowledge of an interface as taught by Zeiger combined with Bayeh fails to suggest or disclose the specific type of interface with the specific characteristics recited in Claim 1 in each servlet of Bayeh. Applicants have demonstrated multiple reasons why the obviousness rejection is not well founded. Only one of the reasons is necessary to overcome the rejection. Applicants request reconsideration and withdrawal of the obviousness rejection of Claim 1.

Claims 8, 10, 13, 18 and 24 stand rejected as obvious for the same or equivalent reasons as Claim 1. Each of Claims 8, 10, 13, 18 and 24 includes limitations similar to that noted above with respect to Claim 1 and so the remarks concerning Claim 1 and the combination of references are directly applicable to the rejection of each of these claims and are incorporated herein by reference. Applicant respectfully requests reconsideration and withdrawal of the obviousness rejection of each of Claims 8, 10, 13, 18 and 24.

Claim 9 depends from Claim 8 and so distinguishes over combination of references for at least the same reasons as Claim 8. Applicants request reconsideration and withdrawal of the obviousness rejection of Claim 9.



Claims 15 and 16 depend from Claim 13 and so distinguish over the various combinations with Bayeh for at least the same reasons as Claim 13. Applicants request reconsideration and withdrawal of the obviousness rejection of each of Claims 15 to 16.

Claim 19 depends from Claim 18 and so distinguishes over combination of references for at least the same reasons as Claim 18. Applicants request reconsideration and withdrawal of the obviousness rejection of Claim 19.

With respect to Claims 3 and 21, the obviousness rejection relies upon the same information as discussed above with respect to Claim 1 and in addition the Quirksmode reference. Applicants respectfully traverse the obviousness rejection of Claim 3. Claims 3 and 21 includes limitations similar to that noted above with respect to Claim 1 and so the remarks concerning Claim 1 and the combination of references are directly applicable to the rejections of Claims 3 and 21 and are incorporated herein by reference. The rejection also stated:

"most browsers at the time of the invention supported a plurality of formats, including at least HTML, text, images and style sheets."

Applicants were not able to find any teaching in the Quirksmode reference that supported this statement and so if the rejection is continued, Applicants respectfully request citation with more specificity to the reference. In addition, the conclusion directly contradicts the primary reference that stated:

. . . Because browsers expect an incoming response to be formatted using HTML, servers generate their response in that format. (Emphasis Added.)

Bayeh, Col. 2, lines 52 to 54

and

This is necessary because browsers, by convention, expect to receive data that has been formatted with HTML.

Bayeh, Col. 11, lines 37, 38.

Further, there has been no citation in either of the primary references to processing that differentiated between three different formats as recited in these claims. Applicants request reconsideration and withdrawal of the obviousness rejection of each of Claims 3 and 21.

Claim 5 depends from Claim 3 and so distinguishes over Bayeh for at least the same reasons as Claim 3. Applicants request reconsideration and withdrawal of the obviousness rejection of Claim 5.

With respect to the obviousness rejection of Claim 20, only Bayeh was relied upon. The rejection ignored explicit claim limitations and reduced the limitations to a gist by stating:

The server is not described by Bayeh and is inherently responsible for basic support of the servlets described by Bayeh. . . . it would have been obvious . . . to separate the tasks already performed Bayeh's server into any number of distinct objects, as it would improve readability of code as well as make maintenance and troubleshooting easier.

Basic support of a servlet fails to suggest or disclose the specific services and factories recited in Claim 20. As discussed above and incorporated herein by reference, Bayeh teaches a comma separated file to implement "servlet aliasing" or "Mime types" in a property file to trigger the chained processing and so expressly teaches away from

a conversion service;

a protocol reader coupled to said conversion service wherein said conversion service sets up said protocol reader to determine a source data format;

a chain factory coupled to said conversion service, wherein said conversion service calls said chain factory with at least said source data format and a target data format;

a service manager coupled to said chain factory and to said partial filter adapter library; and  
a filter registry service

The comma separated list or the properties files were used in Bayeh and not the elements recited in this Claim. Applicants respectfully request reconsideration and withdrawal of the obviousness rejection of Claim 20.

Claims 7, 17, 22 and 23 stand rejected under 35 U.S.C. §103(a) as being unpatentable over the combination of references in view of Garshol, "Free XML Software," (12/15/1999).

Assuming arguendo the combination of references is correct and the Examiner's interpretation of the secondary reference is correct, the additional information cited by the Examiner fails to overcome the basic deficiencies of Bayeh as noted above for the claims upon which each of Claims 7, 17, 22 and 23 depend. Therefore, Applicants request reconsideration and withdrawal of the obviousness rejection of each of Claims 7, 17, 22 and 23.

\\  
\\  
\\  
\\  
\\  
\\  
\\  
\\  
\\

Appl. No. 09/759,742  
Amdt. dated July 26, 2006  
Reply to Office Action of April 26, 2006



Claims 1, 3, 5, 7 to 10, 13, and 15 to 24 remain in the application. Claims 2, 4, 6, 11, 12, and 14 were canceled previously. For the foregoing reasons, Applicant(s) respectfully request allowance of all pending claims. If the Examiner has any questions relating to the above, the Examiner is respectfully requested to telephone the undersigned Attorney for Applicant(s).

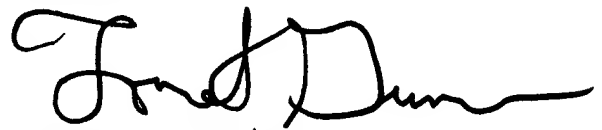
**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on July 26, 2006.

  
Attorney for Applicant(s)

July 26, 2006  
Date of Signature

Respectfully submitted,



Forrest Gunnison  
Attorney for Applicant(s)  
Reg. No. 32,899  
Tel.: (831) 655-0880